DEFECT-TOLERANT AND FAULT-TOLERANT CIRCUIT INTERCONNECTIONS

TECHNICAL FIELD

The present invention relates to nanoscale electronic circuits and, in particular, to defect-tolerant interconnection interfaces with electrically distinguishable signal levels, including interfaces between microelectronic and nanoelectronic circuits used in a combination nanoscale/microscale electronic memory, combination nanoscale/microscale logic circuits, field-programmable gate arrays, and processors.

BACKGROUND OF THE INVENTION

The present invention is related to nanoscale electronic circuit interconnections, including memory devices employing nanowire crossbars and defect-tolerant interconnections with electrically distinguishable signal levels between microelectronic circuit elements and nanowires. Nanowire crossbar technologies, and a variety of nanoscale electronic circuits, are discussed in a number of issued U.S. Patents and filed U.S. Patent Applications, including (1) U.S. Patent No. 6,459,095, entitled "Chemically Synthesized and Assembled Electronic Devices," issued to James R. Heath et al. on October 1, 2002; (2) U.S. Patent 6,314,019, entitled "Molecular Wire Crossbar Interconnect (MWCI) for Signal Routing and Communications," issued to Philip J. Kuekes et al. on November 6, 2001; (3) U.S. Application No. 09/280,045, entitled "Molecular Wire Crossbar Logic (MWCL)," filed on March 29, 1999, in the names of Philip J. Kuekes et al.; (4) U.S. Patent 6,128,214, entitled "Molecular Wire Crossbar Memory," issued to Philip J. Kuekes et al. on October 3, 2000; and (5) U.S. Patent 6,256,767, entitled "Demultiplexer for a Molecular Wire Crossbar Network," issued to Philip J. Kuekes et al. on July 3, 2001, all assigned to the same assignee as the present application.

Nanowire crossbars provide an enormous increase in device density compared with current, photolithography-produced microelectronic and sub-microelectronic circuitry. However, many alignment and spontaneous defect problems need to be overcome to successfully manufacture electronic devices that include nanoelectronic circuits, including nanowire crossbars. A number of techniques and nanowire-crossbar implementations have been designed to overcome these alignment and defect problems, including configurable, or reprogrammable, nanowire-crossbar implementations that allow defects to be detected and circumvented by programming configurations that provide desired functionality without incorporating defective molecular junctions. These techniques are not needed for current microelectronic circuitry produced by photolithographic techniques, because microelectronic circuits can be assumed, in the current discussion, to be essentially perfect or, more precisely, the defect rate in photolithography-produced microelectronic circuits is so far below the current defect rate in nanoscale electronic circuitry that the comparatively very low defect rate in microelectronic circuitry can be ignored.

Nanoscale electronic circuits need to interface to microelectronic circuitry in order to be incorporated within commercially viable and useful electronic devices. The interface between essentially non-defective microelectronic circuitry and nanoelectronic circuitry is problematic. While various correctly functioning nanoelectronic circuits can be configured from defect-prone nanowire crossbars, these nanoscale circuits need to be interconnected with microelectronic circuits. The relatively high manufacturing-defect rate that occurs in fabricating nanoelectronic devices may produce a yield of usable combination nanoscale/microscale circuits too low to be economically viable. However, it is not feasible to extend existing redundancy-based, defect avoidance strategies designed for nanoscale circuits to microelectronic circuitry through nanoscale-to-microscale interfaces, because these redundancy-based techniques depend on an ability to attempt a measurement of each junction in the nanoscale circuits to determine whether or not the junction is defective. Such individual access to junctions within an interconnection interface would require the interconnection interface to be properly functioning in the first place. In other words, the redundancy-based techniques assume defective nanoscale circuitry components, but rely on an ability to address the components through a properly working interconnection interface. Thus, defects in the interconnection interface result in degradation or complete lack of addressability of interconnection interface components. The interconnection interface may be viewed as a type of bootstrap mechanism that allows defect control in a nanoscale circuit, the components of which are addressed through the bootstrap For example, electronic memories based on nanowire-crossbar-implemented memory-element arrays are attractive with respect to size and power consumption, but are not economically viable using current designs and manufacturing strategies, because defects in the interconnect would make large portions of the memory unaddressable, pushing the effective cost/bit of manufacture to a level too high to be competitive.

Another problem that arises in nanoscale electronics is that the separations between ON and OFF voltage or current states may not be sufficiently large to distinguish between ON and OFF states in the presence of noise and imperfect manufacturing tolerances in certain types of circuits, including in groups of nanowires activated by intersecting address signal lines in demultiplexers and other types of circuits. For binary-logic-based circuitry, an easily detected voltage or current separation between "1," or ON, and "0," or OFF, states is needed. In certain types of demultiplexers, for example, one of a large set of nanowires may be addressed by the voltage or current states on a smaller number of address lines that cross the nanowire of interest. If the selected nanowire is designed to have a low, "0," or, synonymously, OFF state, and the unselected nanowires are designed to have a high, "1," or, synonymously, ON state, then the voltage or current difference between the selected nanowire and the lowest voltage or lowest current unselected nanowire must be sufficiently large to be easily detected. Unfortunately, the electronic characteristics of nanowire molecular junctions may be difficult to precisely manufacture, leading to leaky diodes, resistors with a wide variation in resistance, and leaky transistors, in turn leading to undesirably narrow differences between ON and OFF states in addressed nanowires.

These same considerations arise in a variety of different types of systems in which a signal level is classified as belonging to one of a plurality of different, distinguishable classes based on one or more thresholds separating the signal-level classes. To date, correct separation of signals into discrete, distinguishable signal-level classes has been largely accomplished by means of precise manufacturing tolerances. However, more recent classes of systems that discriminate between various types of signals are not amenable to sufficiently precise manufacturing, of which nanoscale electronics is but one example. Additional examples include microfluidics-based systems, which may depend on chemical-signal thresholds for which precise manufacture, detection, and operation may be problematic. Additional examples include microelectromechanical ("MEMS") systems, hybrid electrical systems featuring nanoscale, microscale, and macroscale components, and quantum computing.

For these reasons, designers and manufacturers of nanoscale electronic circuitry, and, in particular, nanoscale electronic memories, have recognized the need for defect-tolerant interconnection within nanoscale circuitry in the interface between microscale and nanoscale circuits. Moreover, designers and manufacturers of nanoscale electronic circuitry have recognized the need for interconnections with distinguishable signal levels between nanowires and address signal lines that select one or a subset of nanowires to have a different voltage or current state than the remaining nanowires.

SUMMARY OF THE INVENTION

One embodiment of the present invention is an electronic memory comprising an array of nanowire crossbars, the nanoscale memory elements within the nanowire crossbars addressed through conventional microelectronic address lines. In order to interconnect the microelectronic address lines with the nanowire crossbars, an address encoding technique is employed to generate a number of redundant address lines to supplement a minimally required set of address signal lines needed to access the nanoscale memory elements. The supplemental address lines allow for unique memory-element addressing in the presence of significant nanowire-to-microelectronic-address-line interconnection defects, provided that the functional interrelationships between the supplemental address lines and the minimally required set of address lines are properly constrained. The functional interrelationships may be mathematically specified using generator matrices originally derived for error-control coding used in data transmission, or may be specified using combinatoric codes or random codes. In the same embodiment, the supplemental address signal lines serve to increase the voltage or current separation between selected nanowires and non-selected nanowires when a particular address is input to the address signal lines in order to select one, or a subset of, the nanowires.

In alternate embodiments, nanocomponent addresses are input to an encoder that outputs corresponding coded addresses by using encoding techniques that increase the average Hamming-distance separation of the coded addresses with respect to the average Hamming-distance separation of the nanocomponent addresses. In other alternate embodiments, nanocomponent addresses input to an encoder are encoded into coded addresses by encoding techniques that increase the minimum Hamming-distance separation of the coded addresses with respect to the minimum Hamming-distance separation of the nanocomponent addresses.

The methods of the present invention are applicable to a wide range of different types of systems in which a signal level is classified as belonging to one of a plurality of different, distinguishable classes based on one or more thresholds separating the signal-level classes. Systems to which the methods of the present invention are applicable include microfluidics-based systems, which may depend on chemical-signal thresholds for which precise manufacture, detection, and operation may be problematic. Additional examples include microelectromechanical ("MEMS") systems, hybrid electrical systems featuring nanoscale, microscale, and macroscale components, and quantum computing.

BRIEF DESCRIPTION OF THE DRAWINGS

Figure 1 shows a high-level diagram of a combined nanoscale/microscale electronic memory device that represents one embodiment of the present invention.

Figure 2 shows an abstract representation of a nanowire-crossbar memory-element subarray within a combined nanoscale/microscale electronic memory device.

Figures 3A-B shows a simple nanoscale-to-microscale interface within a combined nanoscale/microscale electronic memory device.

Figures 4A-D illustrate voltages on the horizontal nanowires resulting from four possible signal patterns input to the internal microelectronic address signal lines in the nanoscale interconnection interface illustrated in Figures 3A-B.

Figure 5 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface.

Figure 6 shows the nanoscale interconnection interface discussed above with reference to Figure 3A-4D with a failed rectifying connection.

Figures 7A-D, using the same illustration conventions of Figures 4A-D, illustrate voltage states of nanowires that arise with each different, two-bit signal pattern input to the address signal lines.

Figure 8 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figure 5, for the defective nanoscale interconnection interface illustrated in Figures 6-7D.

Figure 9 illustrates a nanoscale interconnection interface in which two rectifying interconnections are defective.

Figure 10 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5 and 8, for the defective nanoscale interconnection interface illustrated in Figure 9.

Figure 11 shows a modified nanoscale interconnection interface, similar to the interconnect interfaces illustrated in Figures 3A-4D, that represents one embodiment of the present invention.

Figures 12A-D illustrate nanowire voltage states arising from input of four different, two-bit addresses to the address signal lines of the nanoscale interconnection interface, representing one embodiment of the present invention, shown in Figure 11.

Figure 13 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, and 10, for the defective nanoscale interconnection interface illustrated in Figures 11 – 12D.

Figure 14 shows the modified nanoscale interconnection interface shown in Figure 11, representing one embodiment of the present invention, with a single defective rectifying interconnect.

Figures 15A-D, analogous to Figures 12A-D, show voltage states of nanowires arising from four different, two-bit addresses input to the address signal lines.

Figure 16 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, 10, and 13, for the defective, modified nanoscale interconnection interface illustrated in Figures 14 – 15D.

Figure 17 shows the modified nanoscale interconnection interface first shown in Figure 11 with two defective rectifying connections.

Figure 18 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, 10, 13, and 16, for the defective, modified nanoscale interconnection interface illustrated in Figure 17.

Figure 19 shows a defective, modified nanoscale interconnection interface similar to that shown in Figure 11, , representing one embodiment of the present invention, but having three defective rectifying connections.

Figure 20 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, 10, 13, and 16, for the defective, modified nanoscale interconnection interface illustrated in Figure 17.

Figures 21 and 22 illustrate loss of unique addressability in the modified nanoscale interconnection interface, which involves the loss of two connections from microwires to one nanowire

Figures 23A-B illustrate the comparative robustness of two-bit addresses and three-bit addresses in the nanoscale interconnection interfaces discussed with respect to Figures 3A-22.

Figures 24A-B illustrate address-space topologies of the two-bit, fully occupied address space and a three-bit, sparsely occupied address space.

Figure 25 illustrates a generalized technique for providing parity-check address signal lines via an encoder component in order to increase the defect-tolerance and fault-tolerance of nanoscale interconnection interfaces that represents one embodiment of the present invention.

Figures 26A-B illustrate the derivation of an encoding circuit that adds parity-check address signal lines to an incoming set of address signal lines and derivation of a demultiplexer corresponding to the encoding circuit that both represent embodiments of the present invention.

Figure 27 shows a plot of the normalized expected number of addressable bits per unit area plotted against the defect rate for a 16K-bit cross-point memory using no supplemental address signal lines and using a number of different linear-block codes for specifying different numbers of supplemental address signal lines.

Figure 28 shows plots of an expected addressable percentage of nanowires versus a probability of open connections for a nanoscale interconnection interface having no supplemental address signal lines and nanoscale interconnection interfaces designed using various linear block codes to specify different numbers of supplemental address signal lines.

Figure 29A illustrates a diode-based demultiplexer, similar to demultiplexers illustrated in previous figures that represent embodiments of the present invention.

Figure 29B illustrates a transistor-based demultiplexer, equivalent to the demultiplexer illustrated in Figure 29A, that represents an embodiment of the present invention.

DETAILED DESCRIPTION OF THE INVENTION

Various embodiments of the present invention include a combined nanoscale/microscale electronic memory device and fault-tolerant interconnections between microelectronic circuits and nanoelectronic circuits within the electronic memory. Techniques for determining, in one described approach, nanoelectronic-to-microelectronic interface topologies employ concepts and mathematical techniques developed for error-control coding used in various data-transmission applications. Therefore, in a first subsection, a mathematical description of a number of error-control encoding techniques is provided. In a second subsection, a partially nanoscale electronic memory that represents one embodiment of the present invention is described. In a third subsection, a method for determining the nanoelectronic-to-microelectronic circuit-interface topologies within the electronic memory are described.

Mathematical Description of Selected Error-Control Encoding Techniques

Embodiments of the present invention employ concepts derived from well-known techniques in error-control encoding. An excellent reference for this field is the textbook "Error Control Coding: The Fundamentals and Applications," Lin and Costello, Prentice-Hall, Incorporated, New Jersey, 1983. In this subsection, a brief description of the error-detection and error-correction techniques used in error-control encoding are described. Additional details can be obtained from the above-referenced textbook, or from many other textbooks, papers, and journal articles in this field. The current subsection represents a rather mathematically precise, but concise, description of certain types of error-control encoding techniques. The current invention employs concepts inherent in these error-control encoding techniques for a different purpose. Error-control encoding techniques systematically introduce supplemental bits or symbols into plain-text messages, or encode plain-text messages using a greater number of bits or symbols than absolutely required, in order to provide information in encoded messages to allow for errors arising in storage or transmission to be detected and, in some cases, corrected. One effect of the supplemental or more-than-absolutely-needed bits or symbols is to increase the distance between valid codewords, when codewords are viewed as vectors in a vector space and the distance between codewords is a metric derived from the vector subtraction of the codewords. The current invention employs concepts used in error-control coding to add supplemental address signal lines to increase the distance between valid addresses in order to correspondingly increase the signal separation, e.g. voltage or current, between ON and OFF states of address signal lines and to provide defective-junction tolerance in interface interconnections. Thus, in the current invention, the plain-text and encoded messages of error-control encoding are analogous to input addresses and coded addresses, and the additional or greater-number-than-needed symbols or bits in error-control encoding are analogous to supplemental or a greater-than-absolutely-needed number of address signal lines.

In describing error detection and correction, it is useful to describe the data to be transmitted, stored, and retrieved as one or more messages, where a message μ comprises an ordered sequence of symbols, μ_i, that are elements of a field F. A message μ can be expressed as:

$$\mu=(\mu_0,\mu_1,\dots\mu_{k-1})$$

where $\mu_i \in F$.

The field F is a set that is closed under multiplication and addition, and that includes multiplicative and additive inverses. It is common, in computational error detection and correction, to employ fields comprising a subset of integers with sizes equal to a prime number, with the addition and multiplication operators defined as modulo addition and modulo multiplication. In practice, the binary field is commonly employed. Commonly, the original message is encoded into a message c that also comprises an ordered sequence of elements of the field F, expressed as follows:

$$c=(c_0,c_1,\dots c_{n-1})$$

where $c_i \in F$.

Block encoding techniques encode data in blocks. In this discussion, a block can be viewed as a message μ comprising a fixed number of symbols k that is encoded into a message c comprising an ordered sequence of n symbols. The encoded message c generally contains a greater number of symbols than the original message μ, and therefore n is greater than k. The r extra symbols in the encoded message, where r equals n - k, are used to carry redundant check information to allow for errors that arise during transmission, storage, and retrieval to be detected with an extremely high probability of detection and, in many cases, corrected.

In a linear block code, the 2^k codewords form a k-dimensional subspace of the vector space of all n-tuples over the field F. The Hamming weight of a codeword is the number of non-zero elements in the codeword, and the Hamming distance between two codewords is the number of elements in which the two codewords differ. For example, consider the following two codewords **a** and **b**, assuming elements from the binary field:

$$\mathbf{a} = (1\ 0\ 0\ 1\ 1)$$

$$\mathbf{b} = (1\ 0\ 0\ 0\ 1)$$

The codeword **a** has a Hamming weight of 3, the codeword **b** has a Hamming weight of 2, and the Hamming distance between codewords **a** and **b** is 1, since codewords **a** and **b** differ only in the fourth element. Linear block codes are often designated by a three-element tuple [*n*, *k*, *d*], where *n* is the codeword length, *k* is the message length, or, equivalently, the base-2 logarithm of the number of codewords, and *d* is the minimum Hamming distance between different codewords, equal to the minimal-Hamming-weight, non-zero codeword in the code.

The encoding of data for transmission, storage, and retrieval, and subsequent decoding of the encoded data, can be notationally described as follows, when no errors arise during the transmission, storage, and retrieval of the data:

$$\mu \rightarrow c(s) \rightarrow c(r) \rightarrow \mu$$

where *c(s)* is the encoded message prior to transmission, and *c(r)* is the initially retrieved or received, message. Thus, an initial message μ is encoded to produce encoded message *c(s)* which is then transmitted, stored, or transmitted and stored, and is then subsequently retrieved or received as initially received message *c(r)*. When not corrupted, the initially received message *c(r)* is then decoded to produce the original message μ. As indicated above, when no errors arise, the originally encoded message *c(s)* is equal to the initially received message *c(r)*, and the initially received message *c(r)* is straightforwardly decoded, without error correction, to the original message μ.

When errors arise during the transmission, storage, or retrieval of an encoded message, message encoding and decoding can be expressed as follows:

$$\mu(s) \rightarrow c(s) \rightarrow c(r) \rightarrow \mu(r)$$

Thus, as stated above, the final message μ_r may or may not be equal to the initial message $\mu(s)$, depending on the fidelity of the error detection and error correction techniques employed to encode the original message $\mu(s)$ and decode or reconstruct the initially received message $c(r)$ to produce the final received message $\mu(r)$. Error detection is the process of determining that:

$$c(r) \neq c(s)$$

while error correction is a process that reconstructs the initial, encoded message from a corrupted initially received message:

$$c(r) \rightarrow c(s)$$

The encoding process is a process by which messages, symbolized as μ, are transformed into encoded messages c. Alternatively, a messages μ can be considered to be a word comprising an ordered set of symbols from the alphabet consisting of elements of F, and the encoded messages c can be considered to be a codeword also comprising an ordered set of symbols from the alphabet of elements of F. A word μ can be any ordered combination of k symbols selected from the elements of F, while a codeword c is defined as an ordered sequence of n symbols selected from elements of F via the encoding process:

$$\{c : \mu \rightarrow c\}$$

Linear block encoding techniques encode words of length k by considering the word μ to be a vector in a k-dimensional vector space, and multiplying the vector μ by a generator matrix, as follows:

$$c = \mu \cdot \mathbf{G}$$

Notationally expanding the symbols in the above equation produces either of the following alternative expressions:

$$(c_0, c_1, \ldots, c_{n-1}) = (\mu_0, \mu_1, \ldots, \mu_{k-1}) \begin{pmatrix} g_{00} & g_{01} & g_{02} & \cdots & g_{0,n-1} \\ & \vdots & & \ddots & \vdots \\ g_{k-1,0} & g_{k-1,1} & g_{k-1,2} & \cdots & g_{k-1,n-1} \end{pmatrix}$$

$$(c_0, c_1, \ldots, c_{n-1}) = (\mu_0, \mu_1, \ldots, \mu_{k-1}) \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \cdot \\ \cdot \\ \cdot \\ \mathbf{g}_{k-1} \end{pmatrix}$$

where $g_i = (g_{i,0}, g_{i,1}, g_{i,2} \cdot\cdot\cdot g_{i,n\text{-}1})$.

The generator matrix G for a linear block code can have the form:

$$\mathbf{G}_{k,n} = \begin{pmatrix} p_{0,0} & p_{0,1} & \cdot\cdot\cdot & p_{0,r-1} & 1\ 0\ 0 \ldots 0 \\ p_{1,0} & p_{1,1} & \cdot\cdot\cdot & p_{1,r-1} & 0\ 1\ 0 \ldots 0 \\ . & . & \ldots & & 0\ 0\ 1 \ldots 0 \\ . & . & \ldots & & .\ .\ \ldots\ldots \\ . & . & \ldots & & .\ .\ \ldots\ldots \\ p_{k-1,0} & p_{k-1,1} & \cdot\cdot\cdot & p_{k-1,r-1} & 0\ 0\ 0 \ldots 1 \end{pmatrix}$$

or, alternatively:

$$\mathbf{G}_{k,n} = [\mathbf{P}_{k,r} \mid \mathbf{I}_{k,k}].$$

Thus, the generator matrix G can be placed into a form of a matrix P augmented with a k by k identity matrix $I_{k,k}$. A code generated by a generator in this form is referred to as a "systematic code." When this generator matrix is applied to a word μ, the resulting codeword c has the form:

$$c = (c_0, c_1, \ldots, c_{r-1}, \mu_0, \mu_1, \ldots, \mu_{k-1})$$

where $c_i = \mu_0 p_{0,i} + \mu_1 p_{1,i}, \ldots, \mu_{k-1} p_{k-1,i})$.

Note that, in this discussion, a convention is employed in which the check symbols precede the message symbols. An alternate convention, in which the check symbols follow the message symbols, may also be used, with the parity-check and identity submatrices within the generator matrix interposed to generate codewords conforming to the alternate convention. Thus, in a systematic linear block code, the codewords comprise r parity-check symbols c_i followed by the symbols comprising the original word μ. When no errors arise, the original word, or message μ, occurs in clear-text form within, and is easily extracted from, the corresponding codeword. The parity-check symbols turn out to be linear combinations of the symbols of the original message, or word μ.

One form of a second, useful matrix is the parity-check matrix $H_{r,n}$, defined as:

$$\mathbf{H}_{r,n} = [\mathbf{I}_{r,r} \mid -\mathbf{P}^T]$$

or, equivalently,

$$\mathbf{H}_{r,n}=\begin{pmatrix} 1\ 0\ 0\ldots 0 & -p_{0,0} & -p_{1,0} & -p_{2,0} & \cdots & -p_{k-1,0} \\ 0\ 1\ 0\ldots 0 & -p_{0,1} & -p_{1,1} & -p_{2,1} & \cdots & -p_{k-1,1} \\ 0\ 0\ 1\ldots 0 & -p_{0,2} & -p_{1,2} & -p_{2,2} & \cdots & -p_{k-1,2} \\ .\ .\ .\ .\ .\ .\ .\ . & . & . & \ldots & . \\ 0\ 0\ 0\ldots 1 & -p_{0,r-1} & -p_{1,r-1} & -p_{0,r-1} & \cdots & -p_{k-1,r-1} \end{pmatrix}.$$

The parity-check matrix can be used for systematic error detection and error correction. Error detection and correction involves computing a syndrome S from an initially received or retrieved message *c(r)* as follows:

$$S=(s_0,s_1,\ldots,s_{r-1})=c(r)\cdot\mathbf{H}^T$$

where $\mathbf{H}^T$ is the transpose of the parity-check matrix $H_{r,n}$ expressed as:

$$\mathbf{H}^T=\begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ . & . & . & \ldots & 1 \\ -p_{0,0} & -p_{0,1} & -p_{0,2} & \cdots & -p_{0,r-1} \\ -p_{1,0} & -p_{0,1} & -p_{0,2} & \cdots & -p_{0,r-1} \\ -p_{2,0} & -p_{0,1} & -p_{0,2} & \cdots & -p_{0,r-1} \\ . & . & . & \ldots & . \\ -p_{k-1,0} & -p_{k-1,1} & -p_{k-1,2} & \cdots & -p_{k-1,r-1} \end{pmatrix}.$$

Note that, when a binary field is employed, x = -x, so the minus signs shown above in $\mathbf{H}^T$ are generally not shown.

Hamming codes are linear codes created for error-correction purposes. For any positive integer m greater than or equal to 3, there exists a Hamming code having a codeword length n, a message length k, number of parity-check symbols r, and minimum Hamming distance $d_{\min}$ as follows:

$$\begin{aligned} n&=2^m-1 \\ k&=2^m-m-1 \\ r&=n-k=m \\ d_{\min}&=3 \end{aligned}$$

The parity-check matrix H for a Hamming Code can be expressed as:

$$\mathbf{H} = [\mathbf{I}_m \,|\, \mathbf{Q}]$$

where $\mathbf{I}_m$ is an $m \times m$ identity matrix and the submatrix $\mathbf{Q}$ comprises all $2^m - m - 1$ distinct columns which are m-tuples each having 2 or more non-zero elements. For example, for m = 3, a parity-check matrix for a [7,4,3] linear block Hamming code is

$$\mathbf{H} = \begin{pmatrix} 1\,0\,0\,0\,1\,1\,1 \\ 0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,1\,1\,0\,1\,1 \end{pmatrix}$$

A generator matrix for a Hamming code is given by:

$$\mathbf{G} = [\mathbf{Q}^T \; \mathbf{I}_{2^m - m - 1}]$$

where $\mathbf{Q}^T$ is the transpose of the submartix $\mathbf{Q}$, and $\mathbf{I}_{2^m - m - 1}$ is a $(2^m - m - 1) \times (2^m - m - 1)$ identity matrix. By systematically deleting l columns from the parity-check matrix **H,** a parity-check matrix **H'** for a shortened Hamming code can generally be obtained, with:

$$\begin{aligned} n &= 2^m - l - 1 \\ k &= 2^m - m - l - 1 \\ r &= n - k = m \\ d_{\min} &\geq 3 \end{aligned}$$

As will be discussed, below, one embodiment of the present invention involves employing the above-described error-control coding techniques to a very different problem space, in which, rather than generating codewords of length $k + r$ from messages of length k, interconnections between $k + r$ internal signal lines are generated from addresses input into k input signal lines. In other words, one embodiment of the present invention involves applying error-control coding techniques to addresses within an address space of size 2^k to generate interconnection mappings between address signal lines and up to 2^k signal lines addressed by the 2^k addresses.

Other types of codes are employed to increase the Hamming distance between codewords in various applications. Many of these alternative codes do not have the convenient properties of linear block codes, including easy generation using generator matrices, and the transparent, pass-through feature of linear block codes allowing for the encoded value to be directly read from the code word. For linear block codes, a plain-text message transfers directly to a codeword containing, in addition, parity-check symbols or bits. In other types of codes, the plain-text message is not directly readable in a corresponding codeword. In both cases, codewords contain a greater number of symbols or bits than absolutely needed to enumerate all valid messages to be encoded. In the case of linear block codes, the additional symbols or bits are parity-check symbols or bits that supplement the plain-text symbols or bits, while in the other types of codes, valid messages are distributed throughout a vector space of dimension equal to the codeword size. It should be noted that, in the following descriptions of the present invention, the term "supplemental address lines" refers to either parity-check address signal lines, analogous to parity-check symbols or bits in linear block codes, or to the greater-number-than-absolutely-needed address signal lines, analogous to the greater-number-than-needed symbols or bits in these other types of codes. However, these other codes may have different advantages that provide utility in different applications.

Combinatoric codes provide a straightforward approach to increasing the Hamming distance between codewords. To create a combinatoric code (also known as a "constant-weight code" or an "r-hot code"), one may select combinations of r bits having a fixed number of 1's from a total codeword space of n bits to produce $C_r^n = \frac{n!}{r!(n-r)!}$ codewords of length n. Of course, one can produce a symmetrical code with an identical number of codewords by choosing combinations of r bits having a fixed number of 0's from a total codeword space of n bits. For example, a combinatoric code including $C_r^n = \frac{n!}{r!(n-r)!} = 165$ codewords can be obtained by choosing all possible 11-bit codewords with exactly three bits having the value "1," the codewords provided in the following table:

11100000000	11010000000	11001000000	11000100000	11000010000	11000001000
11000000100	11000000010	11000000001	10110000000	10101000000	10100100000
10100010000	10100001000	10100000100	10100000010	10100000001	10011000000
10010100000	10010010000	10010001000	10010000100	10010000010	10010000001
10001100000	10001010000	10001001000	10001000100	10001000010	10001000001
10000110000	10000101000	10000100100	10000100010	10000100001	10000011000
10000010100	10000010010	10000010001	10000001100	10000001010	10000001001
10000000110	10000000101	10000000011	01110000000	01101000000	01100100000
01100010000	01100001000	01100000100	01100000010	01100000001	01011000000
01010100000	01010010000	01010001000	01010000100	01010000010	01010000001
01001100000	01001010000	01001001000	01001000100	01001000010	01001000001
01000110000	01000101000	01000100100	01000100010	01000100001	01000011000
01000010100	01000010010	01000010001	01000001100	01000001010	01000001001
01000000110	01000000101	01000000011	00111000000	00110100000	00110010000
00110001000	00110000100	00110000010	00110000001	00101100000	00101010000
00101001000	00101000100	00101000010	00101000001	00100110000	00100101000
00100100100	00100100010	00100100001	00100011000	00100010100	00100010010
00100010001	00100001100	00100001010	00100001001	00100000110	00100000101
00100000011	00011100000	00011010000	00011001000	00011000100	00011000010
00011000001	00010110000	00010101000	00010100100	00010100010	00010100001
00010011000	00010010100	00010010010	00010010001	00010001100	00010001010
00010001001	00010000110	00010000101	00010000011	00001110000	00001101000
00001100100	00001100010	00001100001	00001011000	00001010100	00001010010
00001010001	00001001100	00001001010	00001001001	00001000110	00001000101
00001000011	00000111000	00000110100	00000110010	00000110001	00000101100
00000101010	00000101001	00000100110	00000100101	00000100011	00000011100
00000011010	00000011001	00000010110	00000010101	00000010011	00000001110
00000001101	00000001011	00000000111			

Table 1

It is somewhat more complex to encode messages into combinatoric codes, but the logic to do so may be straightforwardly constructed at the logic-circuit level. Combinatoric codes have a guaranteed minimum Hamming distance of 2, and may have significantly better average Hamming distance separations between codewords. For example, in the above $\binom{11}{3}$ code, the average Hamming distance between codewords is 4.39. Combinatoric codes also have an advantage in producing total signal distinguishability within relatively narrow ranges, since these codes have constant weights, where the weight is defined as the number of bits having the value "1."

Another, similar type of code, referred to as a "random" code, is obtained by choosing random codewords of fixed length. For example, one can choose a fixed-length, binary, *n*-bit codeword size, and select a sufficient number of random n-bit binary numbers in order to obtain a desired number of binary codewords 2^k, where $n > Ak$. The greater the value of A, the greater the expected minimum Hamming distance between the codewords. When creating random codes, distance checking can be carried out to reject new codewords that have a Hamming distance less than a minimum value with respect to those codewords already selected, and random codewords having approximately equal numbers of "1" and "0" bits can be used in order to obtain an increased average Hamming distance and increased expected minimum Hamming distance.

Yet another type of code that may be employed in the methods and systems of the present invention is a random linear code. In a random linear code, the generator matrix is randomly generated, under linearity constraints, rather than generated as the combination of a parity-check matrix generated from linear sums of information elements that represent parity-check sums, and an identity matrix. A random linear block code is generally not systematic, but linear.

In general, codes that may be employed in the methods and systems of the present invention may be systematic and linear, systematic and non-linear, non-systematic and linear, or non-systematic and non-linear. For example, if C is a code, and $\mathbf{u}$ is an arbitrary n-vector, then the *coset* $C' = \mathbf{u} + C = \{ \mathbf{u}+\mathbf{c} : \mathbf{c} \in C \}$ is another code with the same distance properties, and hence with the same error correction and defect tolerance capabilities. If C is linear, and $\mathbf{u}$ is non-zero, then C' is non-linear, technically, an affine space. Certain embodiments, discussed below, employ systematic non-linear codes (*see*, for example, Figures 11-12), while others employ systematic linear block codes (*see*, for example, Figures 26A-B). The random codes are generally neither systematic nor linear. Although linear block codes have properties that are attractive in the applications to be discussed below, linear codes, systematic codes, and non-linear, non-systematic codes may also be employed in various embodiments of the present invention.

A Nanoscale Electronic Memory Device That Represents One Embodiment of the Present Invention

In the current subsection, a combined nanoscale/microscale electronic memory device that represents one embodiment of the present invention is discussed. The present invention is not limited in applicability to nanoscale/microscale electronic memory devices, or even to nanoscale/microscale electronic devices, but may also find application in purely nanoscale interconnection interfaces and other nanoscale devices, and may also find application in sub-microscale electronic devices, interconnection interfaces, and memories. In the described embodiment, single nanowires are uniquely addressed by addresses transmitted to intersecting microscale address signal lines. In other circuits, subsets of nanowires may be addressed, rather than single nanowires, by a single address. The present invention is related to general addressing of components within a circuit, and not limited to uniquely addressing single components or signal lines, nor limited to components or signal lines of particular sizes.

Figure 1 shows a high-level diagram of a combined nanoscale/microscale electronic memory device that represents one embodiment of the present invention. The combined nanoscale/microscale electronic memory device 100 interfaces to an external electronic environment through reference voltage signal lines 102 and through two sets of address signal lines 104 and 106. The memory elements within the combined nanoscale/microscale electronic memory device 100 can logically be considered to compose a two-dimensional array, with each memory element specified by a pair of coordinates (x,y), where the x coordinate specifies the row of the two-dimensional array inhabited by the memory element and the y coordinate specifies the column of the two-dimensional array inhabited by the memory element. The set of address signal lines 106 may be used to specify one of 2^p two-dimensional memory-element-array rows, where p is the number of address signal lines in the set of address signal lines 106, and the set of address signal lines 104 specifies one of 2^q columns in the logical, two-dimensional array of memory elements, where q is the number of address signal lines in the set of address signal lines 106. Although the dimensions p and q of the two-dimensional array of memory elements need not be equal, in the following discussion, both dimensions will be assumed to be equal to p, in the interest of notational brevity.

The external sets of address signal lines 104 and 106 are electronically used within the electronic memory device 100 to select a column and row of nanowire-crossbar memory-element subarrays, and therefore a particular nanowire-crossbar memory-element subarray, and to select a particular row or column within a selected nanowire-crossbar array. For example, in one configuration, the upper three address signal lines 108 of the set of address signal lines 106 may specify one of seven horizontal rows 110-116 of nanowire-crossbar subarrays, and the upper three address signal lines 118 of the set of address signal lines 104 may specify one of six vertical columns 120-125 of nanowire crossbar memory-element subarrays. The lower three address signal lines 126 in the set of address signal lines 106 specify a particular row of nanoscale memory elements within a selected nanowire-crossbar memory-element subarray, and the lower three address signal lines 128 in the set of address signal lines 104 specifies a particular column of nanoscale memory elements within a selected nanowire-crossbar memory-element subarray. Note that, in general, a larger number of input address signal lines would be employed to address a larger number of columns and rows of nanowire-crossbar memory-element subarrays than shown in Figure 1, as well as a larger number of nanoscale-memory-element rows and columns within each nanowire-crossbar memory-element subarray. The small number of input address lines shown in Figure 1 is chosen for illustration convenience only.

Each row and column of nanowire-crossbar memory-element subarrays is accessed through an encoder 130-142. The encoder receives, in Figure 1, the high-order three input address signal lines and outputs a larger number of coded address signal lines. For example, encoder 137 receives three address signal lines 144 directly interconnected with the high-order input address signal lines 128, and outputs five coded address signal lines 146. The address signal lines output by an encoder pass through all of the nanowire-crossbar memory-element subarrays within the column or row accessed via the encoder. For example, the five coded address signal lines 146 output by the encoder 137 pass through nanowire-crossbar memory-element subarrays 150-156. Each nanowire-crossbar memory-element subarray is also connected to two reference voltage signal lines. For example, nanowire-crossbar memory-element subarray 150 is connected to the reference voltage through reference-voltage signal lines 158 and 160.

The input address signal lines may be microscale address signal lines, or may have even larger dimensions. The coded address signal lines are generally microelectronic or submicroelectronic signal lines produced by currently available photolithographic techniques. The nanowire-crossbar memory-element subarrays, by contrast, are composed of nanoscale wires, or nanowires. Nanowires have cross-sectional diameters of less than 100 nanometers, while submicroelectronic signal lines have cross-sectional diameters of greater than 100 nanometers. Therefore, there are two nanoscale interconnection interfaces within each nanowire-crossbar memory-element subarray. In general, the circuitry within the encoders 130-142 is significantly more complex than the circuitry within the nanowire-crossbar memory-element subarrays. However, each encoder provides access to an entire row or column of nanowire-crossbar memory-element subarrays, thus amortizing the complexity of the encoders over an often large number of nanowire-crossbar memory-element subarrays. As is discussed below, in great detail, defect and fault-tolerant nanoscale interconnection interfaces are facilitated by the supplemental address signal lines output by each encoder.

Figure 2 shows an abstract representation of a nanowire-crossbar memory-element subarray within a combined nanoscale/microscale electronic memory device. The nanowire-crossbar memory-element subarray 200 is composed of a nanowire crossbar 202 with a similarly sized region 204 of horizontally extending nanowires and a similarly sized region 206 of vertically extending nanowires. A vertical column 208 of microelectronic coded address signal lines passes over the region of horizontally extending nanowires 204, with selective rectifying connections, or nanoscale, molecular-junction diode connections, interconnecting microelectronic internal signal lines with particular horizontal extending nanowires. In alternative embodiments, resistive ohmic connections, semi-conductor gate connections, or other types of physical methods at nanowire junctions that determine the signals on the nanowires 204 may be employed. Similarly, a horizontal set 210 of microelectronic coded address signal lines passes over the region 206 of vertically extending nanowires, with the horizontal microelectronic address signal lines selectively interconnected via rectifying connections, or molecular diodes, to selected vertically extending nanowires. Note that each horizontally and vertically extended nanowire is also connected, through a resistive connection, to a vertical reference-voltage signal line 212 and a horizontal reference-voltage signal line 214, respectively. Each unique pattern of ON and OFF voltages, or, equivalently, HIGH and LOW voltages, on the set of vertical internal microelectronic address signal lines 208 uniquely addresses a particular, horizontally extending nanowire, and places that selected nanowire at a significantly different voltage than the remaining horizontally extending nanowires. Similarly, each different pattern of ON and OFF voltages on the set 210 of horizontal internal microelectronic address signal lines selects a unique vertically extending nanowire, and places that selected vertically extending nanowire at a significantly different voltage than the remaining vertically extending nanowires. The selected horizontally extending nanowire and the selected vertically extending nanowire are interconnected at a single crossover point within the nanowire crossbar, and the molecular junction at that crossover point is placed at a different voltage level than all other molecular junctions within the nanowire crossbar via the pattern of ON and OFF voltages present on the set of vertical and horizontal internal microelectronic signal lines 208 and 210.

Figure 3A shows a simple nanoscale-to-microscale interface within a combined nanoscale/microscale electronic memory device. In the simple interface shown in Figure 3A, two internal microelectronic signal lines 302 and 304 pass over and selectively interconnect with four nanowires 306-309. Each nanowire 306-309 is interconnected to a reference-voltage microelectronic signal line 310 via a resistive interconnection 312-315, respectively. Note that each nanowire has been assigned a two-bit address. Nanowire 306 has the address "00," nanowire 307 has the address "01," nanowire 308 has the address "10," and nanowire 309 has the address "11." Note also that each internal microelectronic signal line 302 and 304 is split into a complementary pair. Thus, for example, internal microelectronic address signal line 302 passes over the nanowires as a pair of microelectronic signal lines 316-317, with the right-hand microelectronic signal line 316 of the pair having the same voltage as the internal microelectronic signal line 302, and the left-hand microelectronic signal line 317 of the pair having a voltage complementary to that of the internal microelectronic address signal line 302. Note that the coded address signal lines are not necessarily split into complementary pairs at each nanowire-crossbar memory-element subarray, but may be split once and pass through an entire row or column of nanowire-crossbar memory-element subarrays. Figure 3B shows the selective rectifying crosspoint connections between the complementary-pair, microelectronic address signal lines 316-317 and 318-319 and the four nanowires 306-309. In Figure 3B, each rectifying connection is indicated by a diagonally directed diode symbol, such as diagonally directed diode symbol 320. Those crosspoints, or intersections, without diagonally directed diode schematics are not electrically interconnected.

It should be noted that the term "coded signal line" refers, in this document, to either a single coded signal line or to a complementary pair of signal lines. In the following descriptions, various formulas and expressions are provided that are related to the number of coded signal lines in particular implementations or used in particular techniques. If, for example, a formula or expression refers to *n* coded signal lines, then *n* refers to the number of independent signal lines or to the number of independent, complementary pairs of signal lines. In certain implementations, single, independent coded signal lines may be employed. In other implementations, including the implementations discussed in this document, complementary pairs of signal are employed.

Figures 4A-D illustrate voltages on the horizontal nanowires resulting from four possible signal patterns input to the internal microelectronic address signal lines in the nanoscale interconnection interface illustrated in Figures 3A-B. The voltage V for a nanowire can be computed from the following formula:

$$\frac{V-V_S}{R_S}+\frac{V-V_A}{R_A}+\frac{V-V_B}{R_B}=0$$

where V_A and V_B are the voltages of the relatively high-voltage-signal-carrying internal address signal lines, at a given point in time, connected to the nanowires of interest, e.g. internal address signal lines 317 and 319 for nanowire 309 at the point in time illustrated in Figure 4A, V_S is the source voltage, the resistance R_S is the resistance of the resistor interconnecting reference-voltage signal line 310 with the nanowire, and the resistances R_A and R_B are the effective resistances of the forward-biased or reverse biased rectifying connections, depending on the voltages on the connected address signal lines, if the nanowire is interconnected with the internal microelectronic address signal lines. In this example, the relationship between the resistance of the resistor R_S, the resistance of a forward-biased rectifying connection R_F and the resistance of reverse-biased rectifying connection $R_{R,}$ are given by the following equation:

$$R_R = 10R_S = 100R_F$$

In Figure 4A, the address-signal-line voltage pattern “00” is input into the internal, microelectronic address signal lines 302 and 304. Thus, the complement microelectronic signal lines 317 and 319 are held at a high voltage V_H while the direct address signal lines of the complementary pairs 316 and 318 are held at essentially 0 volts. The reference-voltage signal line 310 is also held at 0 volts. In Figure 4A, and in many similar figures to follow, the address signal lines held at high voltage are indicated by arrows, such as arrow 322, drawn within the address signal line. Horizontal nanowire 309 receives high voltage input from both complementary address signal lines 317 and 319, resulting in a voltage of 0.95 V_H. Each of the horizontal nanowires 307 and 308 receives a single high voltage input from one of the two complementary address signal lines 317 and 319. Horizontal nanowire 306 receives no high voltage input, and is therefore held at the reference voltage zero by the reference-voltage signal line 310. Thus, the address “00” input to the internal, microelectronic address signal lines 302 and 304 results in selection of nanowire “00” 306 for a low voltage state, and a high voltage state held on the remaining three horizontal nanowires 307-309. Figures 4B-4D illustrate the voltage states of the nanowires resulting from input of the remaining three voltage patterns “01,” “10,” and “11” to the internal microelectronic address signal lines, and demonstrates that a single, unique nanowire labeled with the input address is selected to have a low voltage state in each case. Thus, the interface between the nanowires and the microelectronic address signal lines in the memory-element subarray constitutes a demultiplexer that selects a particular nanowire for each input address. Figure 5 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface.

In Figure 5, the nanowire addresses are shown on a vertical axis 502 and the address signals input to the microelectronic address signal lines are shown on a horizontal axis 504. The table clearly reveals that a single nanowire is held at low voltage for each different address, and that the address input to the internal, microelectronic address signal lines is equal to the address assigned to the selected nanowire.

The microscale-to-nanoscale demultiplexer interface discussed above with reference to Figures 3A-4D is fabricated with rectifying connections, or molecular diodes, interconnecting the address signal lines with the nanowires. In alternative embodiments, non-rectifying, resistive connections may be employed, with resistances substantially less than the resistance of the resistor interconnecting the nanowires with the reference-voltage signal line. When resistive interconnections are employed, rather than diode interconnections, the voltage differences between the selected, low-voltage nanowire and the remaining nanowires may be significantly less when a similar high voltage V_H is applied to selected address signal lines. Moreover, while the difference in the voltages of the non-selected nanowires in the above example is small, 0.05 volts, the voltage states of the non-selected nanowires in a resistive connection implementation span a much larger range. Particularly in the resistive connection implementation, as the number of nanowires interconnected to the interface increases, problems arise in establishing voltage differences that are easily distinguished from one another and from the low voltage state of the selected nanowire.

Figure 6 shows the nanoscale interconnection interface discussed above with reference to Figure 3A-4D with a failed rectifying connection. As shown in Figure 6 by the circular dashed line 602, the rectifying connection interconnecting address signal line 318 with horizontal nanowire 307 is missing, leaving the crosspoint between the address signal line 318 and the nanowire 307 in a permanent open state. As discussed briefly in the background section, the nanoscale interconnections cannot, with current technologies, be as reliably fabricated as can interconnections in microscale electronics fabricated using photolithographic processes. Although it is not possible, presently, to reliably fabricate the selective rectifying connections, it is possible, in general, to fabricate the nanoscale interconnection interface so that failures resulting in permanent open crosspoints occur with much greater frequency than failures that result in a permanent short at a crosspoint. Thus, in the present discussion, defect and fault-tolerant designs and techniques are described with reference to ameliorating the effects of permanent open-type connection failures.

Figures 7A-D, using the same illustration conventions of Figures 4A-D, illustrate voltage states of nanowires that arise with each different, two-bit signal pattern input to the address signal lines. As can be seen in Figure 7D, the lack of the rectifying connection (indicated by dashed circle 602 in Figure 6) results in an inability to uniquely address nanowire "11" 309. Figure 8 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figure 5, for the defective nanoscale interconnection interface illustrated in Figures 6-7D. Comparing Figure 5, showing the nanowire voltage states for the perfectly manufactured interconnection interface, to Figure 8, one can immediately appreciate the deleterious effects arising from the missing rectifying connection. First, as discussed above, nanowire "11" 309 cannot be uniquely addressed, as indicated by the two zero voltage states 802 and 804 in the column 806 corresponding to input signal "11." However, an additional effect may be observed. In Figure 5, there is a descending, right diagonal of 0.95 V_H voltage states, while in Figure 8, voltage state 808 for nanowire "01" with input signal "10" is now 0.90 V_H, rather than 0.95 V_H. Thus, not only has the failure of the rectifying connection prevented both nanowires "01" and "11" from being uniquely addressed, the failed rectifying connection has also lowered the cumulative voltage difference between the selected-nanowire, low-voltage state and the high voltage states of the unselected nanowires.

Figure 9 illustrates a nanoscale interconnection interface in which two rectifying interconnections are defective. The two defective rectifying interconnections are illustrated in Figure 9 by dashed circles 902 and 904. Figure 10 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5 and 8, for the defective nanoscale interconnection interface illustrated in Figure 9. Comparison of Figure 10 to Figures 5 and 8 show that the additional failed rectifying connection (904 in Figure 9) has further degraded the nanoscale interconnection interface. Now, only nanowire "00" can be uniquely addressed. For example, input of the signal pattern "10" selects both nanowires "10" and "11," as indicated by the zero voltage states shown in cells 1002 and 1004. As another example, nanowire "01" is selected both by address signal pattern "01" and by signal pattern "11," as indicated by the zero voltage states in cells 1006 and 1010. Also, there are now only two maximum voltage states shown in cells 1012 and 1014, rather than four maximum voltage states shown in the rightward descending diagonal in Figure 5.

Because the nanoscale rectifying connections cannot be perfectly fabricated, and the because the effects of failed, open-state connections so seriously degrades the ability to uniquely address nanowires using different input addresses, a method for ameliorating the effects of failed nanoscale interconnections has been recognized as necessary in order to produce commercially viable products that include nanoelectronic circuits. Figure 11 shows a modified nanoscale interconnection interface, similar to the interconnect interfaces illustrated in Figures 3A-4D, that represents one embodiment of the present invention. As shown in Figure 11, the modified nanoscale interconnection interface 1100 has four addressable nanowires 1102-1105 with two-bit addresses "00," "01," "10," and "11," as in the interconnection interface shown in Figure 3A. The modified interconnection interface also employs two address signal lines 1106 and 1107 into which two-bit addresses may be input. As in the previously described interconnection interface, shown in Figure 3A, each address signal line 1106 and 1107 is split into a complementary pair, 1108-1109 and 1110-1111, respectively. However, an additional complementary pair of signal lines 1112 and 1113, with signals derived from the signals input to the address signal lines 1106 and 1107, are also selectively interconnected to the nanowires 1102-1105 via rectifying nanoscale connections, such as rectifying nanoscale connection 1114. The complementary signal-line pair 1112-1113, represents a parity-check signal line, derived from the address signal lines 1106-1107 by an inverted XOR logic component 1116. In essence, the parity-check complementary pair 1112-1113 adds an additional, although derived, and therefore not independent, address bit, so that the nanowires 1102-1105 can be considered to have three-bit addresses. In Figure 11, the derived, parity-check address bit is shown parenthesized, as, for example, the parenthesized parity-check address bit 1118 for nanowire 1105. Figures 12A-D illustrate nanowire voltage states arising from input of four different, two-bit addresses to the address signal lines of the nanoscale interconnection interface shown in Figure 11. As can be seen in Figures 12A-D, each different input address uniquely selects the nanowire designated as having that address, as in the case of the nanoscale interconnection interface described above with reference to Figures 3A-4D. Figure 13 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, and 10, for the defective nanoscale interconnection interface illustrated in Figures 11 – 12D. Comparison of Figure 13 to Figure 5 reveals an immediate advantage arising from the presence of the parity-check complementary pair (1112 and 1113 in Figure 11). As can be seen in Figure 13, regardless of the input address, each nanowire either has the low-voltage selected-nanowire state or a single high-voltage state. Thus, all voltages shown in Figure 13 are either 0 or 0.95 V_H, while in Figure 5, the non-selected voltage states range from 0.90 V_H to 0.95 V_H. Thus, the added parity-check complementary signal-line pair decreases the range of voltage states for non-selected nanowires, and increases the voltage separation of selected nanowires from non-selected nanowires.

Figure 14 shows the modified nanoscale interconnection interface shown in Figure 11 with a single defective rectifying interconnect. The defective rectifying interconnect is indicated in Figure 14 by the dashed circle 1402. The same rectifying connection defect in Figure 14 is shown as being defective in Figure 6. Figures 15A-D, analogous to Figures 12A-D, show voltage states of nanowires arising from four different, two-bit addresses input to the address signal lines. As can be appreciated by comparing the voltage states shown in Figures 15A-D to those shown in Figures 12A-D, the presence of the additional rectifying connections between the parity-check complementary signal-line pair (1112-1113 in Figure 11) has significantly increased the robustness of a modified nanoscale interconnection interface in comparison to the interconnection interface illustrated in Figure 3A. Figure 16 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, 10, and 13, for the defective, modified nanoscale interconnection interface illustrated in Figures 14 – 15D. Comparison of Figures 16 and 13 shows that, despite the defective rectifying connection (1402 in Figure 14), each nanowire remains uniquely addressable. In other words, unlike in the case of Figures 8 and 10, a single zero-voltage state appears in each column and row of Figure 16. The only effect of the defective rectifying connection is lowering of the high voltage states 1602 and 1604 for nanowire "01" from 0.95 V_H to 0.90 V_H. Thus, the single defective rectifying connection, in the case of the modified nanoscale interconnection interface, broadens the range of non-selected nanowire high voltage states, but does not result in a loss of addressability.

Figure 17 shows the modified nanoscale interconnection interface first shown in Figure 11 with two defective rectifying connections. The two defective rectifying connections are shown by dashed circles 1702 and 1704 in Figure 17. Figure 18 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, 10, 13, and 16, for the defective, modified nanoscale interconnection interface illustrated in Figure 17. As can be seen in Figure 18, all four nanowires remain uniquely addressable despite two defective rectifying connections.

Figure 19 shows a defective, modified nanoscale interconnection interface similar to that shown in Figure 11, but having three defective rectifying connections. The defective rectifying connections are shown by dashed circles 1902, 1904, and 1906 in Figure 19. Figure 20 shows a table indicating voltage states of nanowires resulting from address signals input to microelectronic address signal lines of a nanoscale interconnection interface, using the same illustration conventions as used in Figures 5, 8, 10, 13, and 16, for the defective, modified nanoscale interconnection interface illustrated in Figure 17. As can be easily observed in Figure 20, all four nanowires remain uniquely addressable despite three defective rectifying connections. In fact, each nanowire can lose one out of three rectifying connections within the modified nanoscale interconnection interface and remain uniquely addressable. Unique addressability is lost only when two or more rectifying connections on a single nanowire are defective. Figures 21 and 22 illustrate loss of unique addressability in the modified nanoscale interconnection interface.

Figures 23A-B illustrate the comparative robustness of two-bit address spaces and three-bit address spaces in the nanoscale interconnection interfaces discussed with respect to Figures 3A-22. On the left-hand side of Figure 23A, four different high-to-low address-bit transformations are shown for the two-bit address spaces. For example, the four-address, two-bit address space 2302 with four valid addresses is transformed into a two-bit address space with only three valid addresses 2304 when the lower bit 2306 of address “11” is, through a defective interconnection, converted from “1” to “0.” In this case, the top two addresses collapse into a single address “10,” as indicated by bifurcating arrows 2308. In all cases illustrated in the left-hand portion of Figure 23A, a single transformation of an address bit from “1” to “0” or from "0" to "1" invariably results in a decrease in the number of valid or unique addresses. By contrast, as illustrated in the right-hand side of figure 23A and in Figure 23B, in a sparsely occupied, three-bit address space that is occupied by only four valid addresses, transformation of a single address bit from “1” to “0” or from "0" to "1" leaves four valid addresses in the three-bit address space. If the address at 2312, for example, is changed from “1” to “0,” the resulting address 2313 is still distinguishable from the other addresses. As shown in Figure 23B, a decrease in the number of unique addresses occurs in the sparsely occupied, three-bit address space only when at least two bits within a single address are changed. For example, a two-bit transformation changes the sparsely occupied three-bit address space 2316 with four unique or valid addresses to a three-bit, sparsely occupied address space 2318 with only 3 valid or unique addresses.

This robustness in addressability despite address-bit defects or conversions can be geometrically represented. Figures 24A-B illustrate address-space topologies of the fully occupied, two-bit address space and a sparsely occupied, three-bit address space. In Figure 24A, each different two-bit address is represented by a disk or node. For example, the two-bit address “01” is represented by node 2402. Edges, such as edge 2404, interconnect nodes, the addresses for which can be interconverted by converting a single bit “1” to “0” or from "0" to "1." Thus, as shown by the dotted arrow 2406, a single bit conversion within a two-bit address in the two-bit, fully occupied address space produces another address within the two-bit, fully occupied address space. Figure 24B illustrates a three-bit, sparsely occupied address space. In Figure 24B, four of the eight addresses within the three-bit address space are occupied, or are valid. The four valid addresses are "001" 2408, "010" 2410, "100" 2412, and "111" 2414. Note that the occupied, or valid addresses, are arranged spatially as the vertices of a tetrahedron, and each valid address is separated from any other valid address by at least two edges. Thus, two address-bit conversions are required to produce a different, valid address from any given address, as indicated by the two dotted arrows 2416 and 2418 in Figure 24B. The concepts illustrated geometrically in Figures 24A-B can be generalized to a sparsely occupied address space of any size with addresses arranged so that a transformation of more than one bit is needed to transform one valid address to another.

Graphical representations similar to those shown in Figure 24A-B are commonly employed to represent the distances between codewords in linear block codes used for error-control coding. Linear block codes are discussed above in the subsection on error-control coding. Although error-control coding is employed for real-time transmission of data through imperfect communications media, and for storage and recovery of data from error-prone mass storage devices, the similarity between the problem spaces arising in ameliorating defective interconnections between nanoelectronic and microelectronic circuits and in data transmission and data storage inspired the rigorous approach of the present invention to designing redundant, parity-check address signal lines, in addition to direct address signal lines. These redundant signal lines complement the direct address signal lines, increasing the robustness of nanoscale interconnection interfaces in the manner that the single additional parity-check complementary signal-line pair, first illustrated in Figure 11, markedly increased the defect tolerance of the nanoscale interconnection interface, shown in Figure 11, with respect to the interconnection interface shown in Figure 3A.

Figure 25 illustrates a generalized technique for providing parity-check address signal lines via an encoder component in order to increase the defect-tolerance and fault-tolerance of nanoscale interconnection interfaces. The encoder 2502 receives *k* incoming address signal lines 2504. The encoder outputs the *k* address signal lines 2506 in addition to *r* additional parity-check signal lines 2508. Thus, the encoder receives *k* input address signal lines and outputs *n* coded address signal lines. The signals output through the *r* parity-check signal lines need to be derived from the signals input through the *k* input address signal lines in such a way as to allow for a robust, *n*-bit address space containing *k* well-distributed and valid addresses. This problem has already been solved by the linear block code error-control encoding techniques discussed above. The *n* output signal lines may be viewed as carrying *n*-bit codewords derived from *k*-bit messages, with the encoder 2502 functionally equivalent to a linear block code encoding engine.

Figures 26A-B illustrate the derivation of an encoding circuit that adds parity-check address signal lines to an incoming set of address signal lines and derivation of a demultiplexer corresponding to the encoding circuit. In Figure 26A, the encoder receives seven external input address signal lines m_1 - m_7 2602. The encoder circuit outputs seven unchanged signal lines 2604 corresponding to the input address signal lines 2602. In addition, the encoder circuit 2600 outputs five additional parity-check coded-address signal lines 2606-2610. Each additional parity-check signal line is derived from a subset of the input address signal lines m_1-m_7 2602 through a multiple-input XOR component, such as XOR component 2612. The encoder circuit 2600 is specified by the generator matrix 2614 of a [12, 7, 4] shortened Hamming code. The product of the vector 2616, having each of the external input address signal lines as elements, and a column from the generator matrix specifies the interconnections between the input address signal lines m_1-m_7 and a single output coded address signal line. For example, multiplication of the input address signal line vector 2616 by the first column 2618 in the generator matrix 2614 results in the single signal line m_1. Thus, the first output coded address signal line should be interconnected only with the first external input address signal line m_1. In a general case, each computed set of interconnections would be input to a multi-input XOR component, but, when only a single input is specified, then it can be exported directly, as is the case with *k* inputs in systematic encoding. As another example, the interconnections of the last output coded address signal line 2610 with the input address signal lines m_1-m_7 is obtained by multiplying the vector 2616 by the last column 2620 in the generator matrix. This multiplication results in a vector containing the input address signal lines m_1, m_2, m_3, m_5, and m_6. Note that these five signal lines are input into the multiple-input XOR component 2622 that produces output signal line 2610. Thus, a large number of different possible encoders with different proportions of parity-check signal lines to input address signal lines can be straightforwardly obtained from the generator matrices of the many different possible linear block codes.

The demultiplexer, or nanoscale interconnection interface, can also be easily configured using the generator matrix 2614 used for designing the encoder. Figure 26B illustrates derivation of a demultiplexer corresponding to the encoding circuit derived in Figure 26A. A matrix **A** 2624 with 7 columns and 128 rows that includes each possible input address can be multiplied by the generator matrix **G** 2626, with 12 columns and 7 rows, to produce a matrix **D** 2628 of codewords with 12 columns and 128 rows. Each codeword, or row, of the matrix **D** corresponds to a coded address that designates a particular nanowire within a crossbar, access to the nanowires of which is provided by the demultiplexer. Thus, each row in the matrix **D** corresponds to a nanowire of the accessed nanowire crossbar, and column in the matrix **D** corresponds to an address signal line output from an encoder. For example, the first two rows of the matrix **D** correspond to crossbar nanowires 2630 and 2631, the first seven columns of matrix **D** correspond to the seven coded address signal lines (see also 2604 in Figure 6A) corresponding to the input address signal lines to the encoder (2602 in Figure 26A), and the final 5 columns of matrix **D** correspond to the five coded address signal lines 2606-2610 (see also Figure 26A) output by the encoder. If an element of a codeword in a row of matrix **D** has the value "1," then the nanowire corresponding to the codeword is interconnected to the address signal line corresponding to the column of matrix **D** in which the element occurs. In other words, the "1" values in matrix **D** correspond to the pattern of interconnections between coded address signal lines and nanowires, while "0" values correspond to the pattern of overlapping, but not interconnected nanowires and address signal lines. In many embodiments, each address signal line shown in Figures 26A-B is implemented as a complementary pair, and so the "1" values in matrix **D** correspond to the pattern of interconnections between non-inverted address signal lines of complementary address-signal-line pairs, while the "0" values in matrix **D** correspond to the pattern of interconnections between the inverted, or complementary, address signal lines of the complementary address-signal-line pairs. The generator matrix for a linear block code can therefore be used to straightforwardly design encoders as well as corresponding demultiplexers, or nanoscale interconnection interfaces. The full **A**, **G**, and **D** matrices for the example of Figures 26A-B are provided, below:

Matrix **A**

```
0 0 0 0 0 0 0
0 0 0 0 0 0 1
0 0 0 0 0 1 0
0 0 0 0 0 1 1
0 0 0 0 1 0 0
0 0 0 0 1 0 1
0 0 0 0 1 1 0
0 0 0 0 1 1 1
0 0 0 1 0 0 0
0 0 0 1 0 0 1
0 0 0 1 0 1 0
0 0 0 1 0 1 1
0 0 0 1 1 0 0
0 0 0 1 1 0 1
0 0 0 1 1 1 0
0 0 0 1 1 1 1
0 0 1 0 0 0 0
0 0 1 0 0 0 1
0 0 1 0 0 1 0
0 0 1 0 0 1 1
0 0 1 0 1 0 0
0 0 1 0 1 0 1
0 0 1 0 1 1 0
0 0 1 0 1 1 1
0 0 1 1 0 0 0
0 0 1 1 0 0 1
0 0 1 1 0 1 0
0 0 1 1 0 1 1
0 0 1 1 1 0 0
0 0 1 1 1 0 1
0 0 1 1 1 1 0
0 0 1 1 1 1 1
0 1 0 0 0 0 0
0 1 0 0 0 0 1
0 1 0 0 0 1 0
0 1 0 0 0 1 1
0 1 0 0 1 0 0
0 1 0 0 1 0 1
0 1 0 0 1 1 0
0 1 0 0 1 1 1
0 1 0 1 0 0 0
0 1 0 1 0 0 1
0 1 0 1 0 1 0
0 1 0 1 0 1 1
0 1 0 1 1 0 0
```

```
0 1 0 1 1 0 1
0 1 0 1 1 1 0
0 1 0 1 1 1 1
0 1 1 0 0 0 0
0 1 1 0 0 0 1
0 1 1 0 0 1 0
0 1 1 0 0 1 1
0 1 1 0 1 0 0
0 1 1 0 1 0 1
0 1 1 0 1 1 0
0 1 1 0 1 1 1
0 1 1 1 0 0 0
0 1 1 1 0 0 1
0 1 1 1 0 1 0
0 1 1 1 0 1 1
0 1 1 1 1 0 0
0 1 1 1 1 0 1
0 1 1 1 1 1 0
0 1 1 1 1 1 1
1 0 0 0 0 0 0
1 0 0 0 0 0 1
1 0 0 0 0 1 0
1 0 0 0 0 1 1
1 0 0 0 1 0 0
1 0 0 0 1 0 1
1 0 0 0 1 1 0
1 0 0 0 1 1 1
1 0 0 1 0 0 0
1 0 0 1 0 0 1
1 0 0 1 0 1 0
1 0 0 1 0 1 1
1 0 0 1 1 0 0
1 0 0 1 1 0 1
1 0 0 1 1 1 0
1 0 0 1 1 1 1
1 0 1 0 0 0 0
1 0 1 0 0 0 1
1 0 1 0 0 1 0
1 0 1 0 0 1 1
1 0 1 0 1 0 0
1 0 1 0 1 0 1
1 0 1 0 1 1 0
1 0 1 0 1 1 1
1 0 1 1 0 0 0
1 0 1 1 0 0 1
1 0 1 1 0 1 0
1 0 1 1 0 1 1
1 0 1 1 1 0 0
1 0 1 1 1 0 1
1 0 1 1 1 1 0
1 0 1 1 1 1 1
1 1 0 0 0 0 0
1 1 0 0 0 0 1
1 1 0 0 0 1 0
1 1 0 0 0 1 1
1 1 0 0 1 0 0
1 1 0 0 1 0 1
1 1 0 0 1 1 0
```

```
1 1 0 0 1 1 1
1 1 0 1 0 0 0
1 1 0 1 0 0 1
1 1 0 1 0 1 0
1 1 0 1 0 1 1
1 1 0 1 1 0 0
1 1 0 1 1 0 1
1 1 0 1 1 1 0
1 1 0 1 1 1 1
1 1 1 0 0 0 0
1 1 1 0 0 0 1
1 1 1 0 0 1 0
1 1 1 0 0 1 1
1 1 1 0 1 0 0
1 1 1 0 1 0 1
1 1 1 0 1 1 0
1 1 1 0 1 1 1
1 1 1 1 0 0 0
1 1 1 1 0 0 1
1 1 1 1 0 1 0
1 1 1 1 0 1 1
1 1 1 1 1 0 0
1 1 1 1 1 0 1
1 1 1 1 1 1 0
1 1 1 1 1 1 1
```

Matrix **G**

```
1 0 0 0 0 0 0 1 1 0 0 1
0 1 0 0 0 0 0 1 0 1 0 1
0 0 1 0 0 0 0 0 1 1 0 1
0 0 0 1 0 0 0 1 1 1 0 0
0 0 0 0 1 0 0 1 0 0 1 1
0 0 0 0 0 1 0 0 1 0 1 1
0 0 0 0 0 0 1 1 1 0 1 0
```

Matrix **D**

```
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 0
0 0 0 0 0 1 0 0 1 0 1 1
0 0 0 0 0 1 1 1 0 0 0 1
0 0 0 0 1 0 0 1 0 0 1 1
0 0 0 0 1 0 1 0 1 0 0 1
0 0 0 0 1 1 0 1 1 0 0 0
0 0 0 0 1 1 1 0 0 0 1 0
0 0 0 1 0 0 0 1 1 1 0 0
0 0 0 1 0 0 1 0 0 1 1 0
0 0 0 1 0 1 0 1 0 1 1 1
0 0 0 1 0 1 1 0 1 1 0 1
0 0 0 1 1 0 0 0 1 1 1 1
0 0 0 1 1 0 1 1 0 1 0 1
```

```
0 0 0 1 1 1 0 0 0 1 0 0
0 0 0 1 1 1 1 1 1 1 1 0
0 0 1 0 0 0 0 0 1 1 0 1
0 0 1 0 0 0 1 1 0 1 1 1
0 0 1 0 0 1 0 0 0 1 1 0
0 0 1 0 0 1 1 1 1 1 0 0
0 0 1 0 1 0 0 1 1 1 1 0
0 0 1 0 1 0 1 0 0 1 0 0
0 0 1 0 1 1 0 1 0 1 0 1
0 0 1 0 1 1 1 0 1 1 1 1
0 0 1 1 0 0 0 1 0 0 0 1
0 0 1 1 0 0 1 0 1 0 1 1
0 0 1 1 0 1 0 1 1 0 1 0
0 0 1 1 0 1 1 0 0 0 0 0
0 0 1 1 1 0 0 0 0 0 1 0
0 0 1 1 1 0 1 1 1 0 0 0
0 0 1 1 1 1 0 0 1 0 0 1
0 0 1 1 1 1 1 1 0 0 1 1
0 1 0 0 0 0 0 1 0 1 0 1
0 1 0 0 0 0 1 0 1 1 1 1
0 1 0 0 0 1 0 1 1 1 1 0
0 1 0 0 0 1 1 0 0 1 0 0
0 1 0 0 1 0 0 0 0 1 1 0
0 1 0 0 1 0 1 1 1 1 0 0
0 1 0 0 1 1 0 0 1 1 0 1
0 1 0 0 1 1 1 1 0 1 1 1
0 1 0 1 0 0 0 0 1 0 0 1
0 1 0 1 0 0 1 1 0 0 1 1
0 1 0 1 0 1 0 0 0 0 1 0
0 1 0 1 0 1 1 1 1 0 0 0
0 1 0 1 1 0 0 1 1 0 1 0
0 1 0 1 1 0 1 0 0 0 0 0
0 1 0 1 1 1 0 1 0 0 0 1
0 1 0 1 1 1 1 0 1 0 1 1
0 1 1 0 0 0 0 1 1 0 0 0
0 1 1 0 0 0 1 0 0 0 1 0
0 1 1 0 0 1 0 1 0 0 1 1
0 1 1 0 0 1 1 0 1 0 0 1
0 1 1 0 1 0 0 0 1 0 1 1
0 1 1 0 1 0 1 1 0 0 0 1
0 1 1 0 1 1 0 0 0 0 0 0
0 1 1 0 1 1 1 1 1 0 1 0
0 1 1 1 0 0 0 0 0 1 0 0
0 1 1 1 0 0 1 1 1 1 1 0
0 1 1 1 0 1 0 0 1 1 1 1
0 1 1 1 0 1 1 1 0 1 0 1
0 1 1 1 1 0 0 1 0 1 1 1
0 1 1 1 1 0 1 0 1 1 0 1
0 1 1 1 1 1 0 1 1 1 0 0
0 1 1 1 1 1 1 0 0 1 1 0
1 0 0 0 0 0 0 1 1 0 0 1
1 0 0 0 0 0 1 0 0 0 1 1
1 0 0 0 0 1 0 1 0 0 1 0
1 0 0 0 0 1 1 0 1 0 0 0
1 0 0 0 1 0 0 0 1 0 1 0
1 0 0 0 1 0 1 1 0 0 0 0
1 0 0 0 1 1 0 0 0 0 0 1
1 0 0 0 1 1 1 1 1 0 1 1
```

```
1 0 0 1 0 0 0 0 0 1 0 1
1 0 0 1 0 0 1 1 1 1 1 1
1 0 0 1 0 1 0 0 1 1 1 0
1 0 0 1 0 1 1 1 0 1 0 0
1 0 0 1 1 0 0 1 0 1 1 0
1 0 0 1 1 0 1 0 1 1 0 0
1 0 0 1 1 1 0 1 1 1 0 1
1 0 0 1 1 1 1 0 0 1 1 1
1 0 1 0 0 0 0 1 0 1 0 0
1 0 1 0 0 0 1 0 1 1 1 0
1 0 1 0 0 1 0 1 1 1 1 1
1 0 1 0 0 1 1 0 0 1 0 1
1 0 1 0 1 0 0 0 0 1 1 1
1 0 1 0 1 0 1 1 1 1 0 1
1 0 1 0 1 1 0 0 1 1 0 0
1 0 1 0 1 1 1 1 0 1 1 0
1 0 1 1 0 0 0 0 1 0 0 0
1 0 1 1 0 0 1 1 0 0 1 0
1 0 1 1 0 1 0 0 0 0 1 1
1 0 1 1 0 1 1 1 1 0 0 1
1 0 1 1 1 0 0 1 1 0 1 1
1 0 1 1 1 0 1 0 0 0 0 1
1 0 1 1 1 1 0 1 0 0 0 0
1 0 1 1 1 1 1 0 1 0 1 0
1 1 0 0 0 0 0 0 1 1 0 0
1 1 0 0 0 0 1 1 0 1 1 0
1 1 0 0 0 1 0 0 0 1 1 1
1 1 0 0 0 1 1 1 1 1 0 1
1 1 0 0 1 0 0 1 1 1 1 1
1 1 0 0 1 0 1 0 0 1 0 1
1 1 0 0 1 1 0 1 0 1 0 0
1 1 0 0 1 1 1 0 1 1 1 0
1 1 0 1 0 0 0 1 0 0 0 0
1 1 0 1 0 0 1 0 1 0 1 0
1 1 0 1 0 1 0 1 1 0 1 1
1 1 0 1 0 1 1 0 0 0 0 1
1 1 0 1 1 0 0 0 0 0 1 1
1 1 0 1 1 0 1 1 1 0 0 1
1 1 0 1 1 1 0 0 1 0 0 0
1 1 0 1 1 1 1 1 0 0 1 0
1 1 1 0 0 0 0 0 0 0 0 1
1 1 1 0 0 0 1 1 1 0 1 1
1 1 1 0 0 1 0 0 1 0 1 0
1 1 1 0 0 1 1 1 0 0 0 0
1 1 1 0 1 0 0 1 0 0 1 0
1 1 1 0 1 0 1 0 1 0 0 0
1 1 1 0 1 1 0 1 1 0 0 1
1 1 1 0 1 1 1 0 0 0 1 1
1 1 1 1 0 0 0 1 1 1 0 1
1 1 1 1 0 0 1 0 0 1 1 1
1 1 1 1 0 1 0 1 0 1 1 0
1 1 1 1 0 1 1 0 1 1 0 0
1 1 1 1 1 0 0 0 1 1 1 0
1 1 1 1 1 0 1 1 0 1 0 0
1 1 1 1 1 1 0 0 0 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1
```

Although linear block codes provide conveniences in the above application, combinatoric, random, and other types of codes, including linear codes, systematic codes, linear, systematic codes, and non-linear, non-systematic codes may also be employed in order to produce increased fault-tolerance, defect-tolerance, and increased ON and OFF state voltage or current separation. Techniques fabricating randomly interconnected address signal lines and addressed nanowires have been disclosed, for example, in U.S. Patent Number 6,256,767. Using these techniques to interconnect a larger number of coded-address signal lines than the number of input address signal lines, where the voltage or current states of the supplemental address signal lines are derived by Boolean logic from the input address signal lines, produces an address space corresponding to a random, fixed length code, with very good expected minimum Hamming distances between addresses. However, a table-driven encoder needs to be used in order to encode input addresses into the coded address space. Encoders somewhat more complex than those used for the above-described linear-block-code-determined address spaces can be used to encode input addresses into coded addresses corresponding to a combinatoric code. As discussed above, combinatoric codes also provide minimum Hamming distances of at least 2, and even better average Hamming distances between codewords, and therefore may provide a reasonably high degree of fault tolerance and ON/OFF state voltage or current separation.

Method for Determining Nanoscale Electronic Interconnection Topologies

As discussed above, the mathematical techniques employed for linear block encoding and decoding can be used for constructing encoders and demultiplexers for interfacing microelectronic address signal lines to nanoelectronic circuits. However, there are many different possible linear block codes, having a variety of different ratios of supplemental (e.g. parity-check) symbols to information symbols. A designer of nanoscale interconnection interfaces needs to therefore consider various tradeoffs that arise when the number of supplemental address signal lines is increased to provide greater and greater levels of fault tolerance. On one hand, increasing the supplemental address signal lines increases the ability to uniquely address nanowires in the presence of nanowire-to-address-signal-line connection defects. However, increasing the number of supplemental address signal lines increases the cost of manufacture and increases the complexity, size, and cost of the encoders. Therefore, a designer or manufacturer of devices that include both nanoelectronic circuits and microelectronic circuits needs to select linear block codes, or other specification means, appropriate for the expected or observed connection defect probability in the devices, balancing the cost of supplemental address signal lines with the degree of increased fault-tolerance and defect-tolerance obtained by adding the supplemental address signal lines.

One way to consider and quantify the costs and advantages associated with adding supplemental address signal lines is to determine a yield for the overall electronic circuit defined as the number of addressable memory bits per unit of chip area. The microelectronic circuits, at least in initial designs, make up the greatest part of a mixed nanoscale and microscale electronic circuit. When the area of the nanoelectronic circuitry is small compared to the total area of the mixed circuit, then the area of the mixed circuit is proportional to n, the number of address signal lines and supplemental address signal lines. The expected number of addressable bits per unit area is then given by the expression:

$$B_{C(k,s)} = \frac{k}{k+r}(P_{C(k,r)})^2$$

where the parenthesized subexpression $P_{C(k,r)}$, referred to below as P_{OK}, is the probability that a single nanowire is addressable when a code $C(k,r)$ with k information bits and r supplemental bits is used.

Another way to consider the costs and benefits of supplemental address signal lines is to calculate the probability that a particular nanowire is addressable, despite the occurrence of interconnection defects. Consider two different nanowire addresses, **c** and **d**, generated as codewords in a linear-block code, as discussed above. The two addresses **c** and **d** will collide, due to defective connections, when address **d** is defective in all positions in which its address bits differ from those of address **c**. The probability of such address collisions is $p^{dist(\mathbf{c},\mathbf{d})}$, where p is the probability of a defective nanowire junction. Assuming that defects are statistically independent, the overall probability that address **c** is not disabled by any other address is given by the following expression:

$$P_{OK2} = \prod_{\mathbf{d}\in C\setminus\{\mathbf{c}\}} (1-p^{dist(\mathbf{c},\mathbf{d})}) = \prod_{i=1}^{n}(1-p^{i})^{W_C(i)}$$

where $W_C(i)$ denotes the number of codewords of Hamming weight i in the linear-block code C. The weight profiles of linear block codes are the number of codewords having each possible Hamming weight in the code. Table 1, below, provides the weight profiles for a number of codes and dominating sets (*see* below), obtained by enumeration:

$[n,k]$	[7,7]		[8,7]		[11,7]		[12,7]	
I	$W_C(i)$	$W_{D(C)}(i)$	$W_C(i)$	$W_{D(C)}(i)$	$W_C(i)$	$W_{D(C)}(i)$	$W_C(I)$	$W_{D(C)}(i)$
0	1	1	1	1	1	1	1	1
1	7	7	0	0	0	0	0	0
2	21	21	28	28	0	0	0	0
3	35	35	0	56	13	13	0	0
4	35	35	70	70	26	130	39	39
5	21	21	0	56	24	462	0	312
6	7	7	28	28	24	462	48	924
7	1	1	0	8	26	330	0	792
8	0	0	1	1	13	165	39	495
9	0	0	0	0	0	55	0	220
10	0	0	0	0	0	11	0	66
11	0	0	0	0	1	1	0	12
12	0	0	0	0	0	0	1	1

Next, one can compute the probability of a nanowire not being disabled. Let **e** be the characteristic vector of an error pattern, where an element of **e** has the value "1" if there is a defective connection corresponding to the position of the element for a nanowire. The set of locations where $\mathbf{e}_i$ has the value "1" is called the support of **e**. The error-pattern vector dominates an n-vector **c** if the support of **e** includes the support of **c**. An error pattern **e** on address zero disables address **c** if and only if **e** dominates *C*. If *D(C)* denotes the set

$D(C) = \{\mathbf{e} \mid \mathbf{e}\,\text{dominates}\,\mathbf{c} \text{ for some } \mathbf{c} \in C\}$, then the probability that **e** acting on address zero disables some other address is equal to $\text{Prob}(\mathbf{e} \in D(C))$. The probability that address zero is disabled because it disables some other address is then given by:

$$P_{OK1} = 1 - \text{Prob}(e \in D(C)) = 1 - \sum_{i=1}^{n} W_{D(C)}(i) p^{i} (1-p)^{n-1}$$

Using the weight profiles in the above expressions P_{OK1} and P_{OK2}, the overall probability of a nanowire remaining enabled is provided by the following expression:

$$P_{OK} = P_{OK1} P_{OK2}$$

Figure 27 shows plots of an expected addressable percentage of nanowires versus a probability of open connections for a nanoscale interconnection interface having no supplemental address signal lines and nanoscale interconnection interfaces designed using various linear block codes to specify different numbers of supplemental address signal lines. In Figure 27, the vertical axis 2702 represents the expected percentage of addressable nanowires, and the horizontal axis 2704 represents the fractions of interconnections between a nanowire and an address signal line that are defective, or open. Note that, as the number of coded address signal lines increases, the expected percentage of addressable nanowires also increases.

Figure 28 shows a plot of the normalized expected number of addressable bits per unit area plotted against the defect rate for a 16K-bit cross-point memory using no supplemental address signal lines and using a number of different linear-block codes for specifying different numbers of supplemental address signal lines. In Figure 28, the normalized expected number of addressable bits per unit area ranges from 0 to 1.0 along the vertical axis 2802. The defect ratio, expressed as the fraction of defects in the demultiplexer, in this case, the fraction of open connections, is plotted along the horizontal axis 2804, ranging from 0 to 0.4. The curves in Figure 28 are labeled with the linear-block code specification for the linear block code used to construct the encoders and demultiplexers. At extremely low fractions of defective connections, below point 2806 of the horizontal axis, the uncoded, or no-supplemental-address-signal-lines implementation, specified as [7,7,1], provides the highest number of addressable bits per area. Between the defective connection fraction specified by points 2806 and 2808, a [8,7,2] linear block code having a single supplemental address signal line produces the highest number of addressable bits per chip area. Between defective connection fractions specified by points 2808 and 2810, a [11,7,3] linear block code providing four supplemental address signal lines provides the highest number of addressable bits per area, and above the defective connection fraction specified by point 2810, a [12,7,4] linear block code providing five supplemental address signal lines and a minimum Hamming distance of four provides the greatest expected number of addressable bits per unit area. Thus, different linear-block codes may be used to specify different numbers and encodings of supplemental address signal lines depending on the expected fraction of defective nanowire-to-address-signal-line connections. One may intentionally choose a larger set of supplemental codes if this dramatically lowers the cost of manufacturing the nanoscale multiplexers, by allowing a larger defect fraction to be tolerated

Both yield and expected percentage of addressable nanowire calculations can be employed, along with cost functions, to compute the manufacturing costs of adding supplemental address signal lines to determine the optimal design for encoders and demultiplexers, or nanoscale interconnection interfaces. The above described methods are but a few of the possible approaches to quantifying the tradeoffs in advantages and disadvantages between different interconnection topologies, numbers of supplemental signal lines, different types of address coding, and other such variations and considerations. These methods are not necessarily constrained to select only one of a number of possible methodologies and topologies, but may also select various combinations of methodologies and topologies that confer greater advantages than can be obtained from a single method and topology.

Although the present invention has been described in terms of a particular embodiment, it is not intended that the invention be limited to this embodiment. Modifications within the spirit of the invention will be apparent to those skilled in the art. For example, in addition to linear-block codes, many other types of error-control encoding techniques may be used to design encoders and demultiplexers for composite nanoscale and microscale electronic devices. The nanoscale interconnection interfaces described above are suitable for composite nanoscale and microscale memory devices, but are generally applicable to many different interfaces between microscale and nanoscale electronic circuitry. The technique of adding supplemental address signal lines to a minimal set of signal lines in order to ameliorate the occurrence of defective connections within a nanoscale interconnection interface has almost limitless variations and applications. For example, while resistor-based and diode-based interconnections are discussed above, the techniques of the present invention may also be applied to transistor-based interconnections, where, due to manufacturing imperfections and deficiencies, transistors may leak appreciable current in OFF states. In transistor-based interconnections, the fault-tolerance, defect-tolerance, and ON/OFF state separation characteristics supplied by supplemental address signal lines are useful to produce good yields of nanoscale devices. Figures 29A-B illustrate a diode-based and equivalent transistor-based demultiplexer, similar to demultiplexers illustrated in previous figures. Note that, in transistor-based circuits, the coded address signal lines gate the addressed signal lines, rather than supply current or voltage directly to the addressed signal lines. Although a nanoscale memory device is discussed, above, it should be noted that the techniques of the present invention may be applied to one-dimensionally-addressed nanowire sets, such as a single set of nanowires addressed by a set of address signal lines. Additionally, the techniques of the present invention may be applied not only to two-dimensional memories and circuits, but also to three-dimensional and higher dimensional circuits and memories. While the embodiments discussed above all involve supplemental address signal lines emanating from an encoder, the present invention, in addition, includes embodiments in which the encoder outputs the same number of coded-address signal lines, or even fewer coded-address signal lines, as the number of address signal lines input to the encoder. In these cases, the average Hamming distance separation of the output addresses is greater than the average Hamming distance of the input addresses, resulting in greater defect and fault tolerance in addressing. The methods of the present invention are applicable to a wide range of different types of systems in which a signal level is classified as belonging to one of a plurality of different, distinguishable classes based on one or more thresholds separating the signal-level classes. Systems to which the methods of the present invention are applicable include microfluidics-based systems, which may depend on chemical-signal thresholds for which precise manufacture, detection, and operation may be problematic. Additional examples include microelectromechanical ("MEMS") systems, hybrid electrical systems featuring nanoscale, microscale, and macroscale components, and quantum computing.

The foregoing description, for purposes of explanation, used specific nomenclature to provide a thorough understanding of the invention. However, it will be apparent to one skilled in the art that the specific details are not required in order to practice the invention. The foregoing descriptions of specific embodiments of the present invention are presented for purpose of illustration and description. They are not intended to be exhaustive or to limit the invention to the precise forms disclosed. Obviously many modifications and variations are possible in view of the above teachings. The embodiments are shown and described in order to best explain the principles of the invention and its practical applications, to thereby enable others skilled in the art to best utilize the invention and various embodiments with various modifications as are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalents: